**ЯЗЫК ПРОГРАММИРОВАНИЯ JAVA**

Горбачёв Никита Андреевич

студент, Алматинский технологический университет

*Казахстан, г. Алмата*

Букенова Индира Нурмуханбетовна

преподаватель, магистр технических наук, Алматинский технологический университет

Казахстан, г. Алмата

Төлеушова Айнур Төлендіқызы

Преподаватель, Алматинский технологический университет

Казахстан, г. Алмата

**JAVA PROGRAMMING LANGUAGE**

***Gorbachev Nikita Andreevich***

*student, Almaty Technological University*

*Kazakhstan, Almaty*

***Bukenova Indira Nurmukhanbetovna***

*Lecturer, Master of Technical Sciences, Almaty Technological University*

*Kazakhstan, Almaty*

***Toleushova Ainur Tolendikyzy***

*Lecturer, Almaty Technological University*

*Kazakhstan, Almaty*

**АННОТАЦИЯ**

В этой статье рассмотрен язык программирования Java. Статья подойдет для начинающих пользователей языка программирования Java и расскажет о его общих чертах. В статье будут рассмотрены типы Java программ, типы данных и операторы языка, средства разработки.

**ANNOTATION**

This article discusses the Java programming language. The article is suitable for novice users of the Java programming language and will tell you about its general features. The article will consider the types of Java programs, data types and language operators, development tools.

**Ключевые слова: Java**; средства разработки; апплеты; Типы данных.

**Keywords: Java; development tools; applets; Data types.**

Эта работа посвящена одному из наиболее перспективных языков программирования ныне Java. Технологию под названием Java можно без преувеличения назвать новаторской с точки зрения разработки программного обеспечения. Каков перспективный и инновационный характер этой технологии? Сегодня создание программного обеспечения - серьезная задача. Трудности связаны со старыми архитектурами машин, рабочими процессами, графическими приложениями и т.д. Быстрое развитие технологий в сочетании с Интернетом еще больше усложняет эту консоль. Компьютеры оценивают типы компьютеров IBM PC, Macintosh, Sun Workstations и другие. Даже на краю IBM-совместимых компьютеров существует несколько платформ, например MS Windows 9x Me XP NT 2000, OS 2, Solaris, различные виды покрытия системы UNIX с графической оболочкой XWindows и т.д. Все эти отдельные структуры одной сети, которые должны работать в целом, обеспечивают уровень информационной безопасности. Под влиянием этих причин уровень требований к программному обеспечению значительно возрастает, поэтому приложения должны быть безопасными, высокопроизводительными, работать в распределенной среде и быть промежуточными по архитектуре. Все эти причины привели к необходимости новой перспективы процесса создания и распространения приложений на различных машинах различных архитектур. Требования к переносимости заставили нас отказаться от традиционного способа создания и доставки двоичных файлов, хранящих собственный код и, следовательно, специфичных для платформы.

Система сборки Java от Sun Microsystems отвечает всем этим требованиям. Java является объектно-ориентированным языком, удобным и надежным в использовании благодаря таким преимуществам, как многозадачность, поддержка интернет-протокола и мультиплатформа. Java - интерпретируемый язык, и каждая программа Java написана для предполагаемой машины, называемой условной машиной Java. Результатом такой компиляции является байт-код Java, который в свою очередь может быть выполнен на любой операционной системе, при условии, что существует исполнитель Java, который интерпретирует байт-код в фактический машинный код конкретной системы.

однако такая универсальность этой схемы вызывает несовершенство ресурсоемких компьютерных ресурсов. Так как Java-программы не хранят машинного кода и при их запуске включается Java-исполняющая система, их производительность явно ниже, чем у обычных программ, скомпилированных, например, на языке программирования Си, этот дефект со временем становится менее заметным, в результате повышенной вычислительной мощности компьютерных систем. Язык Java является предметно-ориентированным и поставляется с довольно впечатляющей библиотекой классов. Библиотеки классов Java значительно упрощают разработку приложений, предоставляя программисту мощные инструменты для отладки распространенных проблем. Следовательно, программист сможет уделять больше внимания выходу присоединенных задач, а не такой, как, например, система динамических массивов, взаимодействующих с операционной системой или реализующих компоненты пользовательского интерфейса. Целью предоставленной работы является описание Java, анализ технологии создания и дальнейшего использования программ на этом языке, рассмотрение программных упражнений, показывающих все вышеперечисленные преимущества этого языка. Типы программ Java.

Программы, разработанные на языке программирования Java, можно разделить на две широкие категории в зависимости от их назначения и функциональности: автономные программы (назовем их Java-приложениями), которые свободно работают на локальном компьютере. Апплеты (апплеты) функционируют в Интернете.

Ява в настоящее время поддержана всеми крупнейшими компьютерными платформами. Автономное приложение, разработанное, чтобы бежать самостоятельно, создано и выполнено на местной машине, управляющей Явской системой времени выполнения. Ява достаточно хороша для написания приложений, которые могли также быть написаны в C, C Основной, Дельфи или любой другой язык программирования.

Апплетами, которые делают этот язык популярным, являются своего рода JAVA-приложения, которые объяснены Явской виртуальной машиной, построенной в почти многие сегодняшние браузеры. Каждый апплет - небольшая программа, которая динамично загружена по сети с веб-сервера, когда страница HTML открыта в браузере, который содержит источник для апплета, а также изображения, звука или элемента анимации. Главная специфика апплетов - то, что они - реалистические программы и не только другой том файла, чтобы сохранить мультфильмы или любую другую информацию. Апплет просто не играет тот же сценарий, но отвечает на пользовательское поведение и может динамично изменить его поведение. С апплетами Вы можете сделать страницы веб-сервера динамичными и интерактивными. Апплеты позволяют тяжелую местную обработку данных, приобретенных от веб-сервера или введенных пользователем в клавиатуре. Чтобы улучшить производительность апплетов в браузерах, Собранное Выставление счета (МОНЕТА В ПЯТЬ ЦЕНТОВ) используется на лету. Когда апплет загружен впервые, его кодекс переведен на обычную выполнимую программу, которая сохранена на диске и начата. В результате полная поспешность выполнения Явского апплета несколько раз увеличивается. Из соображений безопасности у апплетов (в отличие от обычных JAVA-приложений) нет доступа к файловой системе локального компьютера. Они могут приобрести все данные за обработку исключительно от веб-сервера. границы подготовки выполнимой программы, Явский исходный файл - текстовый файл, у которого есть одно или несколько определений класса. Явский переводчик предполагает, что первоначальные тексты программ сохранены в файлах с Явскими расширениями. bytecode, произведенный во время процесса перевода для каждого класса, написан в отдельном файле продукции с тем же именем как расширение класса и название класса. Это - класс файлов, которые хранят Bytecode, интерпретируемый Явской системой Времени выполнения в машинный код определенной системы. В первую очередь, Ява Bytecode загружена в систему во время выполнения погрузчиком класса. Погрузчик класса ответственен за погрузку всех классов, должен был запустить приложение. bytecode впоследствии проверен свидетельством Bytecode на отсутствие операций, которые могли нарушить безопасность системы или привести к чрезмерной ситуации. Важно отметить, что Bytecode, классный погрузчик и свидетельство не делают предположений о происхождении кодексов, которые они получают от местной файловой системы или от другого континента. Свидетельство гарантирует, что любой код, который закончил тест, может использоваться переводчиком без риска повреждения его (переводчик), а именно:

Переполнение стека или «исчерпание» не могут произойти

Параметры настройки по байтовым машинам определяют требуемый тип

Доступ к полям и технологиям объектов не нарушает заявленные в классе правила (публичные, частные, защищенные)

После аналогичного теста безопасности байтовый код анализируется в машинный код и работает для выполнения интерпретатором. Кроме того, классы, приобретенные локально (достойный абсолютный доверительный управляющий), и классы, передаваемые по сети из других глобальных (и потенциально враждебных) классов, находятся в разнообразных пространствах имен. При разрешении ссылки на класс он выглядит первым в локальном пространстве. Это предотвращает переопределение «внешними» кодами одного из базовых классов в системе. Кроме того, в процесс интерпретации включаются срочные библиотеки (файлы с расширением JAR). Весь описанный процесс выполнения Java-программ показан на рис.

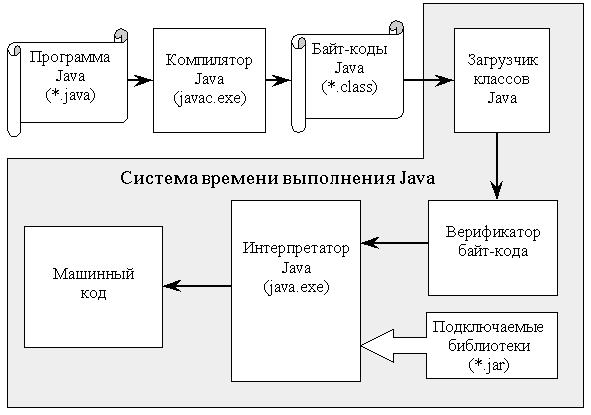


Рис. 1. Процесс создания работающего Java-приложения

Примитивные типы в Java не являются объектно-ориентированными, они похожи на примитивные типы в большинстве традиционных языков программирования. Java имеет восемь простых типов: - Byte, Short, INT, Long, Char, Float, Double и Boolean. Их можно разделить на четыре группы:

К ним относятся байты, короткие, int и длинные типы. Эти типы предназначены для подписывания целых чисел.

Плавающие и двойные типы с плавающей запятой. Они служат для представления чисел, имеющих дробную часть.

Символьный тип. Этот тип предназначен для представления элементов таблицы символов, таких как буквы или цифры.

Логический тип. Это специальный тип, используемый для представления логических значений.

В Java, в отличие от некоторых других языков, отсутствует автоматическое кастинг типа. Несоответствие типов приводит не к появлению предупреждения во время преобразования, а к появлению сообщения об ошибке. Для каждого типа строго определены допустимые наборы значений и разрешенные операции.

Основная система классов Java

Полная система Java включает готовый набор библиотек, наиболее часто используемыми из которых являются следующие пакеты:

Java.lang - основной набор взглядов, отражённых в самом языке. Этот пакет обязательно включен в любое приложение. Содержит описания классов объектов и классов, а также поддержку многопоточности, исключений, оберток для базовых типов, а также некоторых фундаментальных классов.

java.io - произвольные потоки и файлы доступа. Аналог системной библиотеки ввода-вывода системы Unix. В библиотеке классов Java.io осуществляется сбор данных для ввода и вывода, связанных с потоками. С помощью этих классов можно работать не только с потоками байтов, но и с другими типами потоков данных, такими как номера INT или текстовые строки.

Java - поддержка сетевого доступа (сокеты, Telnet, URL). Язык программирования Java разрабатывался с расчётом, что он будет использоваться для создания сетевых приложений. Поэтому было бы странно, если бы среда разработки приложений Java не включала библиотеку классов для сетей. Библиотека классов Java предназначена только для этого. Он содержит классы, с помощью которых можно работать с URL-адресами Universal Network, передавать данные с помощью сокета TCP и UDP и выполнять различные операции с IP-адресами. Эта библиотека также содержит классы для выполнения двоичных преобразований данных в текстовый формат, который часто необходим.

java.util - классы контейнеров (словарь, хэш-таблица, стек) и некоторые другие утилиты. Кодирование и декодирование. КЛАССЫ ДАТЫ И ВРЕМЕНИ. Библиотека классов Java.util очень полезна при компиляции приложений, поскольку имеет классы для создания структур, таких как динамические массивы, стеки и словари. Существуют классы для работы с генератором псевдодома, для разбора строк на составные элементы (токены), для работы с календарной датой и временем.

Java.awt - AWT - аббревиатура от Abstract Window Toolkit (Toolkit для работы с абстрактными окнами). Содержит основные компоненты интерфейса, такие как события, цвета и шрифты. Для создания пользовательского интерфейса Java-апплеты могут и должны использовать библиотеку классов Java.awt. Классы, входящие в библиотеку Java.awt, позволяют создавать пользовательский интерфейс независимо от платформы, на которой выполняется Java-апплет. Можно создавать обычные панели Windows и Dialog, кнопки, переключатели, списки, меню, представления, однострочные и многострочные поля ввода текста.

Java.applet - библиотека классов Java.applet инкапсулирует поведение Java-апплетов. Для создания апплета требуется класс апплета, расположенный в этой библиотеке классов. Кроме того, библиотека классов Java.applet определяет интерфейсы для подключения апплетов к содержащим их документам и классы для воспроизведения звуковых файлов.

**Список литературы:**

1. Майкл Эферган Java: справочник .- QUE Corporation, 1997, Издательство "Питер Ком", 1998
2. Джон Родли Создание Java-апплетов .- The Coriolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.",1996

Материалы интернет-сайтов:

<www.sun.ru>

<https://ru.wikipedia.org/wiki/Java>

<https://clck.ru/epkNz>